

What is claimed is:

- 1 1. A method of determining a variable to receive a value change and a value for
2 the variable as part of a local search solution to an integer programming problem
3 having polynomial terms of at least second order comprising the steps of:
 - 4 a. selecting an unsatisfied constraint;
 - 5 b. creating stores for allowable changes of value for the variables in the
6 unsatisfied constraint;
 - 7 c. parsing through the unsatisfied constraint by term and for each variable in
8 the term updating the stores with a change in the term for each of the
9 allowable changes of the value while maintaining other variables constant; and
 - 10 d. choosing the variable to receive the value change and the value for the
11 variable based upon the store which meets an improvement criterion.
- 1 2. The method of claim 1 wherein the variables comprise a binary variable.
- 1 3. The method of claim 2 wherein all of the variables comprise binary variables.
- 1 4. The method of claim 3 wherein each of the variables is associated with no
2 more than a single store.
- 1 5. The method of claim 3 wherein the step of choosing the variable to receive the
2 value change and the value for the variable comprises choosing the variables to
3 receive the value change based upon which of the stores meets the improvement
4 criterion and flipping the value of the variable selected.
- 1 6. The method of claim 1 wherein the allowable values for at least one of the
2 variables comprise integer values.
- 1 7. The method of claim 6 wherein the integer values comprise a subset of
2 possible integer values.
- 1 8. The method of claim 7 wherein the subset of the possible integer values
2 comprises the integer values near an existing value for the variable.

1 9. The method of claim 1 wherein the allowable changes in the variables do not
2 include tabu values.

1 10. The method of claim 1 wherein the improvement criterion comprises the store
2 which improves the unsatisfied constraint and which improves an overall solution
3 at least as much as other stores.

1 11. The method of claim 1 wherein the improvement criterion comprises the store
2 which is not tabu, which improves the unsatisfied constraint, and which improves
3 an overall solution at least as much as other stores which are not tabu.

1 12. The method of claim 1 further comprising the step of determining that none of
2 the stores improve an overall solution.

1 13. The method of claim 12 wherein the improvement criterion comprises the
2 store which improves the unsatisfied constraint at least as much as other stores.

1 14. The method of claim 12 wherein the improvement criterion comprises a
2 random selection of one of the stores which improves the unsatisfied constraint.

1 15. The method of claim 1 wherein the improvement criterion comprises
2 improvement criteria.

1 16. The method of claim 1 further comprising the steps of:

2 a. defining a problem model which comprises constraints, wherein the
3 constraints comprise hard constraints and a soft constraint;

4 b. randomly initializing values for the variables, thereby forming assigned
5 values for the variables, wherein the assigned values are applied to the
6 constraints to determine the unsatisfied constraints;

7 c. changing the value of the variable according to the improvement criterion,
8 thereby modifying the assigned values to form new assigned values; and

9 d. comparing the constraints to optimality criteria and if the optimality
10 criteria are met, outputting the new assigned values as a near optimum

11 solution.

1 17. The method of claim 16 wherein, if the optimality criteria are not met, further
2 comprising determining whether to perform an additional iteration and if so,
3 returning to step c, wherein the new assigned values become the assigned values.

1 18. The method of claim 17 wherein, if another iteration is not to be performed,
2 further comprising determining whether to reinitialize the variables and if so,
3 returning to step b.

1 19. The method of claim 18 wherein, if the variables are not to be reinitialized,
2 outputting a no solution found message.

1 20. A method of determining a binary variable to receive a value change as part of
2 a solution to an integer programming problem having polynomial terms of at least
3 second order comprising the steps of:

- 4 a. selecting an unsatisfied constraint;
- 5 b. creating a store for each binary variable in the unsatisfied constraint;
- 6 c. parsing through the unsatisfied constraint by term and for each binary
7 variable in the term updating the store for the binary variable with a change in
8 the term due to flipping a value of the binary variable while maintaining other
9 variables constant; and
- 10 d. choosing the binary variable to receive the value change based upon the
11 store which meets an improvement criterion.

1 21. A method of determining a variable to receive a value change and a value for
2 the variable as part of a local search solution to an integer programming problem
3 having polynomial terms of at least second order comprising the steps of:

- 4 a. selecting an unsatisfied constraint;
- 5 b. creating stores for allowable changes of value for the variables in the
6 unsatisfied constraint;
- 7 c. parsing through the unsatisfied constraint by term and for each variable in
8 the term which is encountered for a first time updating the stores with a
9 change in the unsatisfied constraint for each of the allowable changes of the

10 value while maintaining other variables constant; and

11 d. choosing the variable to receive the value change and the value for the
12 variable based upon the store which meets an improvement criterion.

1 22. A computer readable memory comprising computer code for directing a
2 computer to make a determination of a variable to receive a value change and a
3 value for the variable as part of a solution to an integer programming problem
4 having polynomial terms of at least second order, the determination of the variable
5 to change and the value comprising the steps of:

6 a. selecting an unsatisfied constraint;

7 b. creating stores in memory for allowable changes of value for the variables
8 in the unsatisfied constraint;

9 c. parsing through the unsatisfied constraint by term and for each variable in
10 the term updating the stores with a change in the term for each of the
11 allowable changes of the value while maintaining other variables constant; and

12 d. choosing the variable to receive the value change and the value for the
13 variable based upon the store which meets an improvement criterion.

1 23. The computer readable memory of claim 22 wherein the variables comprise a
2 binary variable.

1 24. The computer readable memory of claim 23 wherein all of the variables
2 comprise binary variables.

1 25. The computer readable memory of claim 24 wherein each of the variables is
2 associated with no more than a single store.

1 26. The computer readable memory of claim 25 wherein the step of choosing the
2 variable to receive the value change and the value for the variable comprises
3 choosing the variable to receive the value change based upon which of the stores
4 meets the improvement criterion and flipping the value of the variable selected.

1 27. The computer readable memory of claim 22 wherein the allowable values for
2 at least one of the variables comprise integer values.

- 1 28. The computer readable memory of claim 27 wherein the integer values
2 comprise a subset of possible integer values.
- 1 29. The computer readable memory of claim 28 wherein the subset of the possible
2 integer values comprises the integer values near an existing value for the variable.
- 1 30. The computer readable memory of claim 22 wherein the allowable changes in
2 the variables do not include tabu values.
- 1 31. The computer readable memory of claim 22 wherein the improvement
2 criterion comprises the store which improves the unsatisfied constraint and
3 improves an overall solution at least as much as other stores.
- 1 32. The computer readable memory of claim 22 wherein the improvement
2 criterion comprises the store which is not tabu, which improves the unsatisfied
3 constraint, and which improves an overall solution at least as much as other stores
4 which are not tabu.
- 1 33. The computer readable memory of claim 22 further comprising the step of
2 determining that none of the stores improve an overall solution.
- 1 34. The computer readable memory of claim 33 wherein the improvement
2 criterion comprises the store which improves the unsatisfied constraint at least as
3 much as other stores.
- 1 35. The computer readable memory of claim 33 wherein the improvement
2 criterion comprises a random selection of one of the stores which improves the
3 unsatisfied constraint.
- 1 36. The computer readable memory of claim 22 wherein the improvement
2 criterion comprises improvement criteria.
- 1 37. The computer readable memory of claim 22 further comprising the steps of:

- 2 a. defining a problem model which comprises constraints, wherein the
- 3 constraints comprise hard constraints and a soft constraint;
- 4 b. randomly initializing values for the variables, thereby forming assigned
- 5 values for the variables, wherein the assigned values are applied to the
- 6 constraints to determine the unsatisfied constraints;
- 7 c. changing the value of the variable according to the improvement criterion,
- 8 thereby modifying the assigned values to form new assigned values;
- 9 d. comparing the constraints to optimality criteria and if the optimality
- 10 criteria are met, outputting the new assigned values as a near optimum
- 11 solution.

1 38. The computer readable memory of claim 37 wherein, if the optimality criteria
2 are not met, further comprising determining whether to perform an additional
3 iteration and if so, returning to step c, wherein the new assigned values become
4 the assigned values.

1 39. The computer readable memory of claim 38 wherein, if another iteration is not
2 to be performed, further comprising determining whether to reinitialize the
3 variables and if so, returning to step b.

1 40. The computer readable memory of claim 39 wherein, if the variables are not to
2 be reinitialized, outputting a no solution found message.

1 41. A computer readable memory comprising computer code for directing a
2 computer to make a determination of a binary variable to receive a value change
3 as part of a solution to an integer programming problem having polynomial terms
4 of at least second order, the determination of the binary variable comprising the
5 steps of:

- 6 a. selecting an unsatisfied constraint;
- 7 b. creating a store in memory for each binary variable in the unsatisfied
- 8 constraint;
- 9 c. parsing through the unsatisfied constraint by term and for each binary
- 10 variable in the term updating the store for the binary variable with a change in
- 11 the term due to flipping a value of the binary variable while maintaining other

- 12 variables constant; and
- 13 d. choosing the binary variable to receive the value change based upon the
- 14 store which meets an improvement criterion.

- 1 42. A computer readable memory comprising computer code for directing a
- 2 computer to make a determination of a variable to receive a value change and a
- 3 value for the variable as part of a solution to an integer programming problem
- 4 having polynomial terms of at least second order, the determination of the variable
- 5 to change and the value comprising the steps of:
- 6 a. selecting an unsatisfied constraint;
- 7 b. creating stores in memory for allowable changes of value for the variables
- 8 in the unsatisfied constraint;
- 9 c. parsing through the unsatisfied constraint by term and for each variable in
- 10 the term which is encountered for a first time updating the stores with a
- 11 change in the unsatisfied constraint for each of the allowable changes of the
- 12 value while maintaining other variables constant; and
- 13 d. choosing the variable to receive the value change and the value for the
- 14 variable based upon the store which meets an improvement criterion.